

# Go语言在游戏行业的机会

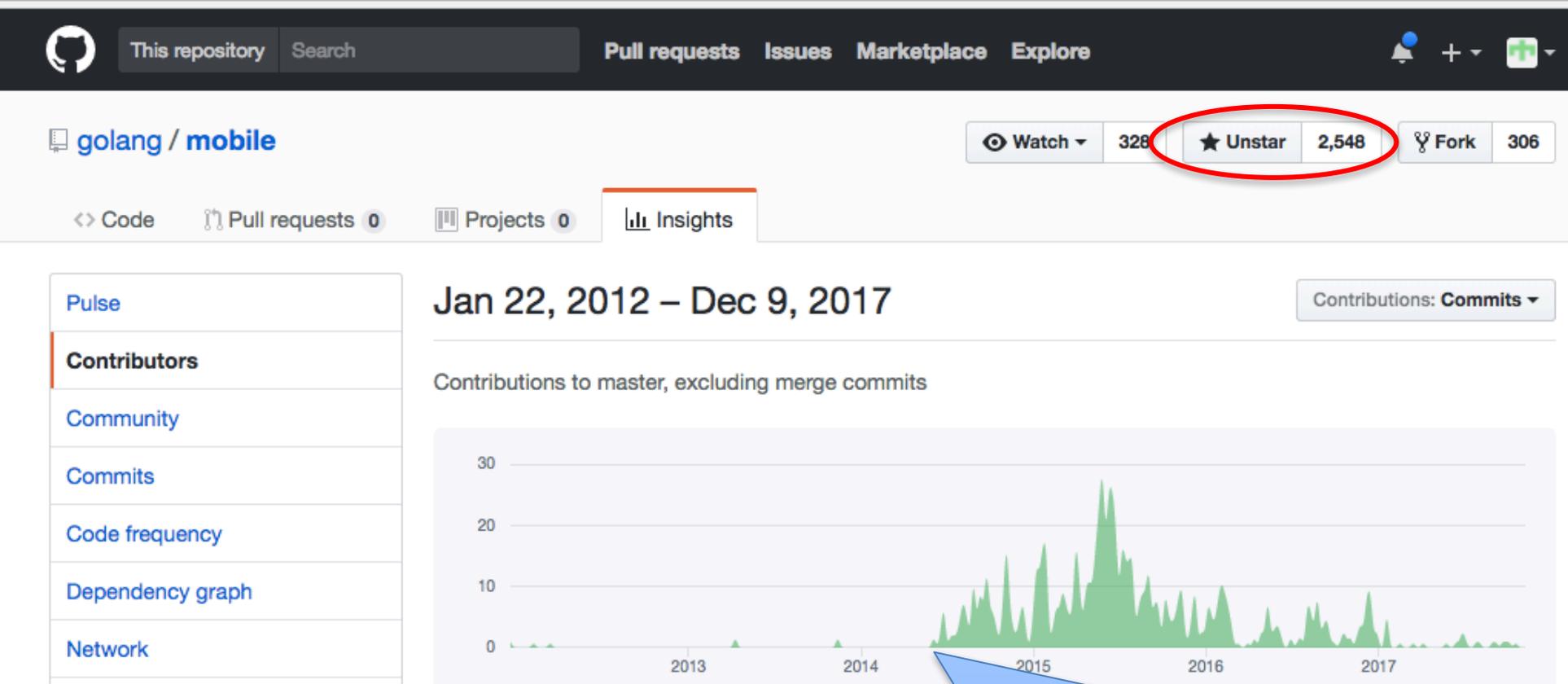
## -- 谈谈Go的桌面侧战场

许式伟

2017-12-10

# Go @mobile

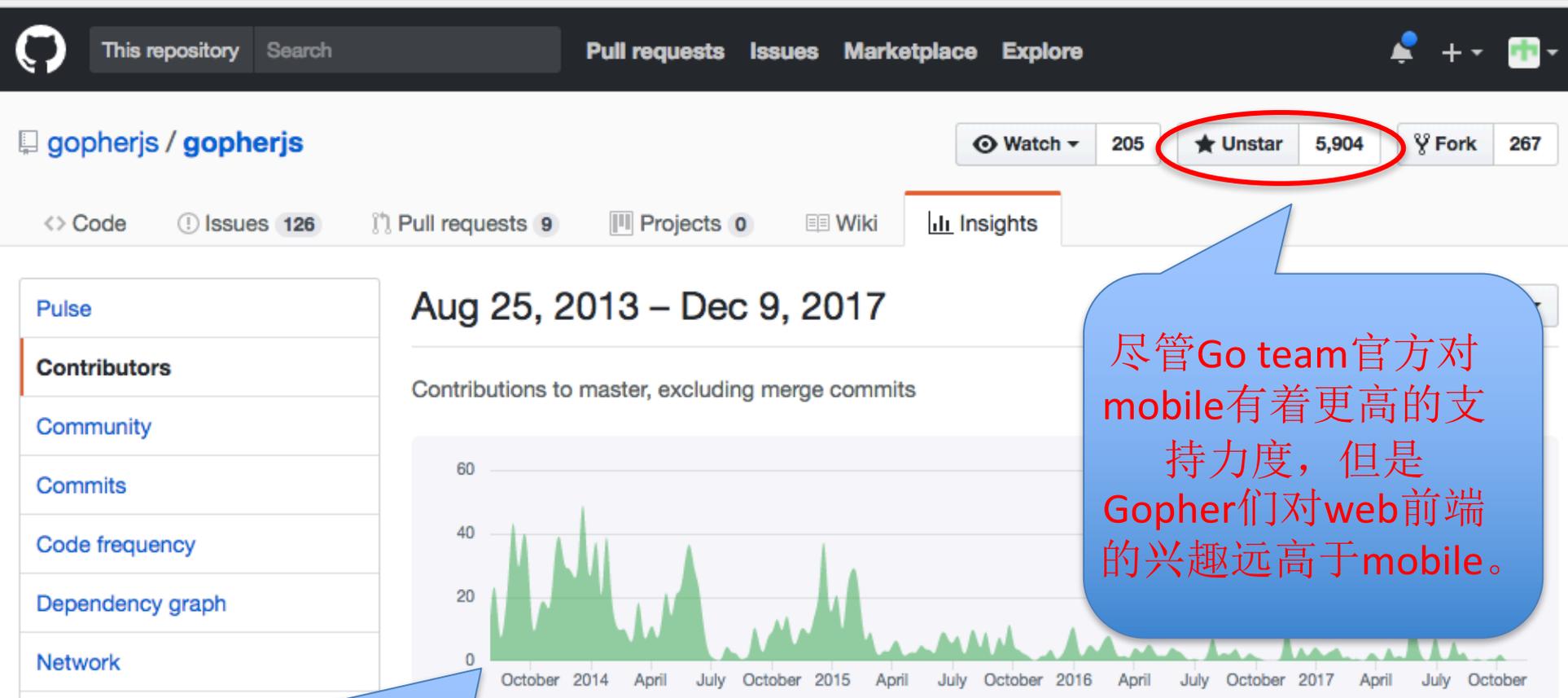
[US] | <https://github.com/golang/mobile/graphs/contributors?from=2012-01-22&to=2017-12-09&type=c>



对mobile的支持开始于2012年初，但实际在2014年中旬才开始活跃

# Go @web

https://github.com/gopherjs/gopherjs/graphs/contributors



尽管Go team官方对mobile有着更高的支持力度，但是Gopher们对web前端的兴趣远高于mobile。

Gopher们和JavaScript程序员一样，有着一个语言统一前后端的梦想。并且这个梦想在2013年8月份就开始了！

# Go语言写桌面程序？

- 很多人在尝试
- 两个流派
  - 通用UI框架派
  - 侧重游戏行业，垂直发展
- Go team选择后者，重点考虑游戏行业

# Why 游戏行业？

- 市场空间巨大
- UI 样式需求个性化
  - 往往桌面OS的标准UI控件被弃用
- 所以
  - 需求相对收敛，容易满足

# Go写游戏的可行性？

- GUI: OpenGL
  - 支持pc (Windows/Mac/Linux/FreeBSD)
  - 支持mobile (Android/iOS)
  - 支持web (基于WebGL)
- 桌面平台
  - 支持pc (通过OpenGL)
  - 支持mobile (通过Go mobile)
  - 支持web (通过GopherJS)
- 所以：一次编写，到处可用！

# Go写页游？

- WebGL
- GopherJS
  - 还可以调用JavaScript框架
    - WebGL: [github.com/gopherjs/webgl](https://github.com/gopherjs/webgl)
    - jQuery: [github.com/gopherjs/jquery](https://github.com/gopherjs/jquery)
    - Websocket: [github.com/gopherjs/websocket](https://github.com/gopherjs/websocket)
    - ...

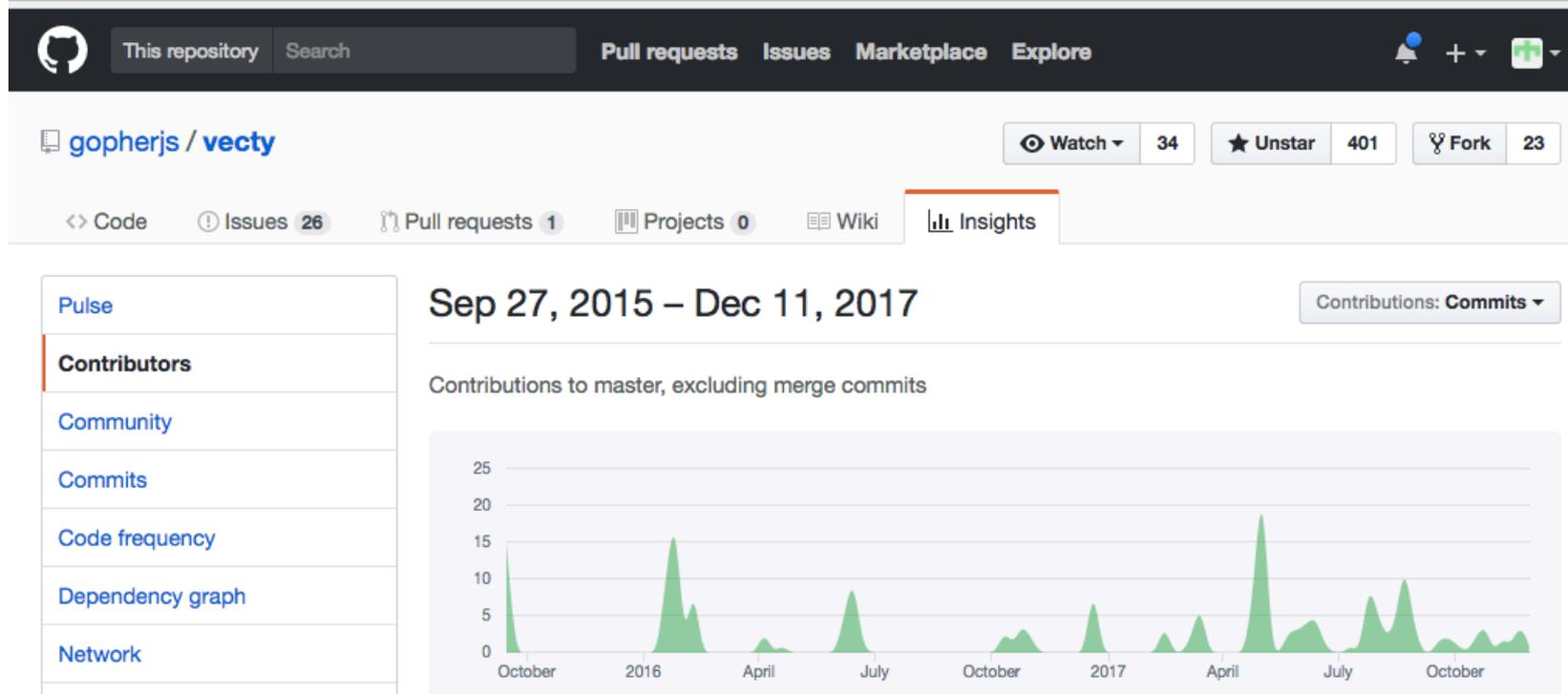
# 页游DEMO

- 俄罗斯方块
- 渲染效率
- 钢琴
- 声音
- 文本

# Go对Web的支持

- 已经成熟，应用面不局限于页游
  - Go对Web支持的成熟度远高于mobile

JS | <https://github.com/gopherjs/vecty/graphs/contributors>



# Go写跨平台游戏？

- 同时支持
  - Windows/Mac/Linux/FreeBSD
  - Android/iOS
  - Web
- 实践一下？

# 用Go做跨平台的Scratch!

- Scratch 是什么?
  - Scratch 是一门语言
  - Scratch 是一门面向儿童编程教学的语言
  - Scratch 是一门教儿童编写游戏的编程教学语言

Nov 2017	Nov 2016	Change	Programming Language	Ratings	
1	1		Java	13.231%	-5.52%
2	2		C	9.293%	+0.09%
3	3		C++	5.343%	-0.07%
4	5	▲	Python	4.482%	+0.91%
5	4	▼	C#	3.012%	-0.65%
6	8	▲	JavaScript	2.972%	+0.27%
7	6	▼	Visual Basic .NET	2.909%	-0.26%
8	7	▼	PHP	1.897%	-1.23%
9	16	▲▲	Delphi/Object Pascal	1.891%	-0.11%
10	9	▼	Assembly language	1.882%	-0.02%
11	19	▲▲	R	1.871%	-0.04%
12	15	▲	MATLAB	1.861%	-0.03%
13	14	▲	Ruby	1.851%	-0.02%
14	13	▼	Go	1.841%	-0.03%
15	10	▼▼	Perl	1.562%	-0.80%
16	26	▲▲	Scratch	1.550%	+0.47%
17	17		Visual Basic	1.489%	-0.43%
18	20	▲	PL/SQL	1.453%	-0.06%
19	11	▼▼	Objective-C	1.412%	-0.83%
20	12	▼	Swift	1.389%	-0.65%

波澜不惊，低调进入前20名的儿童编程教学语言Scratch



# 为何用Go重写Scratch?

- 因为热爱Scratch
- 因为Scratch有一些遗憾
  - Scratch 2.x - <https://github.com/LLK/scratch-flash>
    - 基于Flash编写，只能跑在PC上
  - Scratch 3.0 - <https://github.com/LLK/scratch-gui>
    - 基于Google Blockly + React
    - 已经可用，但还在紧锣密鼓开发中
      - 和 Scratch 2.x 目前不兼容 (但 Scratch 2.x 用户太多了! )

# 目标

- 做一个兼容 Scratch 2.x 的跨PC、移动、Web多终端平台的 Scratch 脚本执行器。

# 挑战1：脚本支持

- Scratch 2.x 的脚本是个json
  - 从 parser 角度来说无难度，从 executor 角度相比其他脚本难度相同
- Scratch 2.x 的脚本是“单线程伪并行的”
  - 类似 JavaScript
  - 不能是串行，脚本语义需要依赖并行能力
  - 不能真并行，因为不同Scratch脚本之间可能共享变量，且Scratch并没有mutex语法，如果真并行会导致系统逻辑错乱

# 挑战2：多行文本排版与显示

- TrueType字体的支持
- 小型排版系统，要考虑行禁则（例如：标点不能在行首，英文单词不能分拆到两行显示）

# 挑战3： 音频播放

- 要支持常见音频格式
- 要支持混音（可以同时播放多个声音，应当要有多个人同时说话的效果）

# 挑战4：SVG格式的支持

- Scratch 2.x内建的精灵(sprite)都是矢量格式的，基于SVG格式
- SVG有着非常复杂的指令集，完整支持等价于写一套GDI Canvas
  - 当前Go社区并无成熟的SVG渲染的项目

# 挑战5：复杂图形

- 当然，假设有SVG支持的前提下，这个只是SVG的一条指令



# 挑战6：碰撞检测

- 这基本上和游戏类程序最基础的需求：两个精灵(sprite)的碰撞检测

# 结果

- 耗时半个月
- 可正确执行我本人写的所有Scratch教程
  - 对话: `weather-conversation.sb2`
    - 音频播放、多行文本排版与显示、复杂图形
  - 五子棋: `five-chess.sb2`
  - 围棋: `go-chess.sb2`
  - 国际象棋: `chess.sb2`
    - 协程(共享全局变量的处理)、复杂脚本
  - 吃蛋糕: `eat-cake.sb2`
    - 碰撞检测

# 跨平台之争： React vs. Go

- React + React Native
  - 优势：倾向于将Web搬到Native(Mobile)
  - 劣势：技术栈有点复杂，性能相对低
  - 场景：比较通用，基于WebGL也可以开发游戏，但React框架对开发游戏的难度基本没有减负
- Go + GopherJS
  - 优势：技术栈极简，上手容易，性能极好
  - 劣势：还较为小众
  - 场景：比较垂直，对游戏有较完整的支持

# 结论

- Go 的桌面侧战场刚刚开启，尽管还很早期，但是值得期待
- 依赖 Go 带来的研发效率提升，可以一定程度消除由于早期而导致社区资源少的不足
- 建议适当关注，择机介入

# 参与方式

- 用Go写游戏，全平台支持
- 用Go写Web应用，替代React等框架